CS5122 DFM Final Project

111062684 林鼎勳

A. Introduction

In this project, students are asked to insert the dummy fills in the layout while considering the density requirements and the spacing between conductors.

B. Command

The Makefile in the submission provides two fundamental commands: `$make` and `$make clean`. The command `$make` will compile my code and generate an executable file named Fill_Insertion. The command `$make clean` will delete the executable file.

C. Algorithm

To solve this problem, we first to choose a grid-based or gridless-based method. The grid-based method means we divide the layout into grids and insert the dummy grid by grid. This method seems easy to implement, however, we may have a problem in the place where the conductors are already too close but it still not met the density requirements. On the other hand, the gridless-based have higher flexibility, but it seems harder than the grid-based method. In my code, I use a grid-based method to solve this problem.

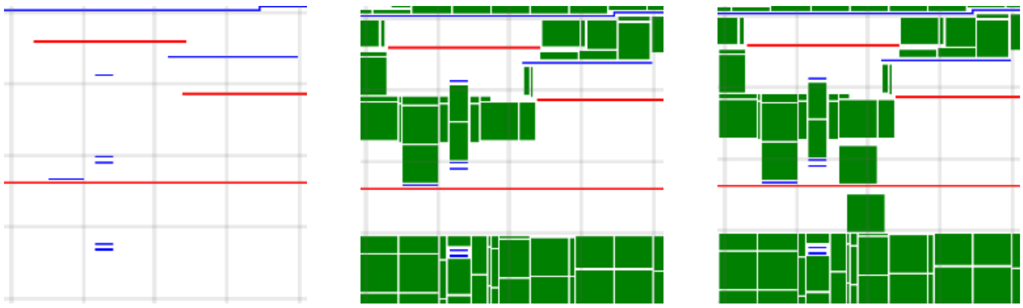
The overall algorithm can be divided into three parts: Grid Creation, Initial Insertion, and Density Refinement.

- Grid Creation: In this part, I first divided the layout into grids. At the same time, I collect the information on conductors and critical nets and put it into the corresponding grid. Therefore, I can get the surrounding information when I insert the dummy fills grid by grid.
- Initial Insertion: In this part, I start to insert dummy fills in *non-critical* areas, and to save time and effort, I also put some fake-inserted dummy fills in the layout. Noted that the fake-inserted dummy fills will not be put into the layout immediately, we just put them as candidates.
- Density Refinement: In the last part, we start calculating the density of each moving window. If the density of a window does not meet the requirement, we will select the fake-inserted dummy fills ordered by the area from large to small and put them into the layout. However, this cannot have all moving windows meet the requirements of density. Here we proposed a moving grid method to tackle it. We will gradually move the already existing grid in any direction in order to produce some grids which are totally empty. That is,

after empty grids exist, we can continually insert dummy fills into the layout until the density of each moving window is above the lower bound.

D. Results

Here are some layouts of benchmark 3 layer 3 at different stages. I use blue color to represent the ordinary conductors, red color to represent the critical nets, and green color to represent the dummy fills. The left-hand side is the initial layout, and it's obvious the density is far away from the lower bound. The middle one is the result after Initial Insertion. All the existing dummy fills are not in the critical region. The right-hand side is the final result. Two additional dummy fills are inserted to meet the requirement of density.



E. Conclusion

In this project, I learned lots of tools that can help to visualize the result like python and matplotlib package. Besides, it also took me many times to find a good and friendly way to handle the 2D problem.